L Number	Hits	Search Text	DB	Time stamp
1	20	(((switch$3) same (crossbar$1 or crosspoint$1)) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/12/22 16:28
2	3	(simultaneous near3 access$3) and ((((switch$3) same (crossbar$1 or crosspoint$1)) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/12/22 16:13
3	4	(simultaneous$2 near3 transfer$5) and ((((switch$3) same (crossbar$1 or crosspoint$1)) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/12/22 16:13
4	4	((simultaneous$2 near3 transfer$5) and ((((switch$3) same (crossbar$1 or crosspoint$1)) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.)) not ((simultaneous near3 access$3) and ((((switch$3) same (crossbar$1 or crosspoint$1)) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/12/22 16:13
5	26	((crossbar$1 or crosspoint$1) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/12/22 16:17
6	6	(((crossbar$1 or crosspoint$1) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.) not ((((switch$3) same (crossbar$1 or crosspoint$1)) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/12/22 16:22
7	4	(multiplexer$1 same switch$3) and (((crossbar$1 or crosspoint$1) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/12/22 16:23
8	93	(((switch$3) same (crossbar$1 or crosspoint$1)) same (arbit$3 or arbitrat$3)) and (multiplexer$1 same switch$3)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/12/22 16:27
9	4	((((switch$3) same (crossbar$1 or crosspoint$1)) same (arbit$3 or arbitrat$3)) and (multiplexer$1 same switch$3)) and 710/316,317.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/12/22 16:28

09/847,991

US-PAT-NO: 6636933

DOCUMENT-IDENTIFIER: US 6636933 B1

TITLE: Data storage system having crossbar switch with multi-staged routing

DATE-ISSUED: October 21, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
MacLellan; Christopher S.	Walpole	MA	N/A	N/A
Walton; John K.	Mendon	MA	N/A	N/A

US-CL-CURRENT: 710/317, 711/114 , 711/152 , 714/5

ABSTRACT:

A memory system having a backplane with a plurality of receiving slots. Each one of the slots has electrical contacts for providing an indication of Such one of the slots. Each one of the slots has a different slot indication. A plurality of memory boards is provided. Each one of the memory boards is plugged into a corresponding one of the slots. Each one of such boards is coupled to the electrical contacts in the corresponding one of the slots to provide a slot signal indicative of the slot indication provided by the electrical contacts. Each one of such boards has: a memory array region; and a switching network for transferring information between a port of the switching network and a memory on such memory boards The transfer is initiated by a director coupled to such port. The director designates a selected one of the plurality of memory boards. The director provides to the switching network a "tag" indicating such designated one of one of the plurality of memory boards having the memory involved in the requested transfer. The switch network includes a memory board checker, for comparing the slot signal with the "tag" for indicating whether the memory board receiving the "tag" is the director designated one of the plurality of memory boards.

12 Claims, 47 Drawing figures

Exemplary Claim Number: 5

Number of Drawing Sheets: 43

---------- KWIC ---------

Detailed Description Text - DETX (20):

Referring now to FIG. 7, an exemplary one of the director boards 190.sub.1 -190.sub.8, 210.sub.1 -210.sub.8, here director board 190.sub.1 is shown to include directors 180.sub.1, 180.sub.3, 180.sub.5 and 180.sub.7. An exemplary one of the directors 180.sub.1 -180.sub.4, here director 180.sub.1 is shown in detail to include the data pipe 316, the message engine/CPU controller 314, the RAM 312, and the CPU 310 all coupled to the CPU interface bus 317, as shown. The exemplary director 180.sub.1 also includes: a local cache memory 319 (which is coupled to the CPU 310); the **crossbar switch** 318; and, the **crossbar switch** 320, described briefly above in connection with FIGS. 5 and 6. The data pipe 316 includes a protocol translator 400, a quad port RAM 402 and a quad port RAM controller 404 arranged as shown. Briefly, the protocol translator 400 converts between the protocol of the host computer 120, in the case of a front-end director 180.sub.1 -180.sub.32, (and between the protocol used by the

disk drives in bank 140 in the case of a back-end director 200.sub.1 -200.sub.32) and the protocol between the directors 180.sub.1 -180.sub.3, 200.sub.1 -200.sub.32 and the global memory 220 (FIG. 2). More particularly, the protocol used the host computer 120 may, for example, be fiber channel, SCSI, ESCON or FICON, for example, as determined by the manufacture of the host computer 120 while the protocol used internal to the system interface 160 (FIG. 2) may be selected by the manufacturer of the interface 160. The quad port RAM 402 is a FIFO controlled by controller 404 because the rate data coming into the RAM 402 may be different from the rate data leaving the RAM 402. The RAM 402 has four ports, each adapted to handle an 18 bit digital word. Here, the protocol translator 400 produces 36 bit digital words for the system interface 160 (FIG. 2) protocol, one 18 bit portion of the word is coupled to one of a pair of the ports of the quad port RAM 402 and the other 18 bit portion of the word is coupled to the other one of the pair of the ports of the quad port RAM 402. The quad port RAM has a pair of ports 402A, 402B, each one of to ports 402A, 402B being adapted to handle an 18 bit digital word. Each one of the ports 402A, 402B is independently controllable and has independent, but **arbitrated,** access to the memory array within the RAM 402. Data is transferred between the ports 402A, 402B and the cache memory 220 (FIG. 2) through the **crossbar switch** 318, as shown.

Detailed Description Text - DETX (47):

More particularly, assume for example that information at upper port 5006.sub.4 (FIGS. 9A, 9B and 9C) of **crossbar switch** 5004.sub.4 is to be transferred to memory array region R.sub.1. Referring to FIG. 10 a negotiation, i.e., **arbitration,** must be made by lower port interface W as a result of a request made by the upper port interface section D of **crossbar switch** 5004.sub.4 to section interface W thereof. When interface section W is available to satisfy such request, (i.e., not satisfying request from other one of the upper port interface sections A-C) interface W issues a grant to upper interface section D.

Current US Original Classification - CCOR (1):

710/317

US-PAT-NO: 6314487

DOCUMENT-IDENTIFIER: US 6314487 B1

TITLE: Adaptive routing controller of a crossbar core module used in a crossbar routing switch

DATE-ISSUED: November 6, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Hahn; Jong Seok	Taejon-shi	N/A	N/A	KR
Sim; Won Sae	Taejon-shi	N/A	N/A	KR
Hahn; Woo Jong	Taejon-shi	N/A	N/A	KR
Yoon; Suk Han	Taejon-shi	N/A	N/A	KR

US-CL-CURRENT: 710/317, 370/351 , 370/389 , 710/306

ABSTRACT:

The present invention relates to a routing control apparatus for performing a round robin arbitration and an adaptive routing control. The present invention relates to a routing controller for performing an arbitration and a routing control which are nucleus functions of the crossbar routing switch and, in particular, to a normal routing controller unit for performing a priority based round robin arbitration and an adaptive routing controller unit for performing an adaptive routing control by adding an adaptive routing switch logic to the normal routing controller.

12 Claims, 5 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 4

---------- KWIC ---------

Abstract Text - ABTX (1):

The present invention relates to a routing control apparatus for performing a round robin arbitration and an adaptive routing control. The present invention relates to a routing controller for performing an arbitration and a routing control which are nucleus functions of the crossbar routing switch and, in particular, to a normal routing controller unit for performing a priority based round robin arbitration and an adaptive routing controller unit for performing an adaptive routing control by adding an adaptive routing switch logic to the normal routing controller.

Brief Summary Text - BSTX (8):

A multipath torus switching apparatus invented by Olnowich and Williams is a switching device of Wormhole routing having small delay time in the interconnection network of torus mesh type(refer to H. T. Olnowich and A. R. Williams, . . . Multipath torus switching apparatus", EP 0588104A2, December 1994). The switching device is invented to determine a routing control port by adding a multipath establishment function to an existing torus switch. The

switching device consists of four single routing controller and a multipath controller and selects an output port by using a connection command of packet. However, the switching device does not suggest a scheme of arbitration method, and has different packet type and routing control method. Although the switching device has a multipath control function similar to the adaptive routing control, since the configuration and method thereof are different, it can not be properly used to a crossbar routing switch.

Brief Summary Text - BSTX (11):

Most of conventional inventions are switching devices or appropriate arbitration methods for the multi-stage interconnection network or the mesh type interconnection network, and in addition, the adaptive routing control methods have many different methods and constitutions, therefore, most of conventional inventions can not provide routing controller adapted to the crossbar routing switch.

Brief Summary Text - BSTX (13):

Therefore, the object of the present invention is to provide a priority based round robin arbitration method for offering fairness and preventing starvation in a routing controller which performs the arbitration and routing control function of the crossbar routing switch, and to provide a routing control apparatus for performing the round robin arbitration and adaptive routing control by inventing a Normal Routing Control Unit which performs the priority based round robin arbitration method and by adding an Adaptive Routing Switch Logic to two the Normal Routing Control Units.

Brief Summary Text - BSTX (14):

The present invention is to provide a crossbar routing switch having a crossbar core module, a plurality of input control module and a plurality of output control module, comprises: a plurality of normal routing control logic for performing a priority based round robin arbitration and controlling an internal logic; an adaptive routing switch logic for receiving a signal of said normal routing control logic and a ready signal and outputting a plurality of corresponding arbitration selection sequential signals, status signals and arbitration selection combinational signals; and a plurality of mask registers for performing the priority based round robin arbitration, wherein said plurality mask registers include a plurality of normal transfer mask registers for connecting, bit by bit, ten bits of the normal arbitration request signal and ten bits driven from said normal routing control logic to the inputs of AND gates and for providing outputs back to said normal routing control logic, and a plurality of emergency transfer mask registers for connecting, bit by bit, ten bits of the emergency arbitration request signal and ten bits driven from said normal routing control logic to the inputs of AND gates and for providing outputs back to said normal routing control logic.

Detailed Description Text - DETX (16):

FIG. 3 shows exterior interface signals of a normal routing controller corresponding tothe normal routing controller units 105a to 105h in the crossbar core module 101 of FIG. 1. The arbitration request units 104a to 104j in the crossbar core module 104 of FIG. 1 interpret the packet tag of FIG. 2 and drive a normal arbitration request signal ReqArb[9 . . . 0] or an emergency arbitration request signal ReqEmg[9 . . . 0] to the corresponding normal routing controller unit 105a to 105h and the corresponding adaptive routing controller unit 106. The input controller modules 102a to 102j of FIG. 1 drive a tail signal tail[9 . . . 0] denoting the end portion of a packet to the corresponding normal routing controller unit 105a to 105h and the corresponding adaptive routing controller unit 106 by using a packet valid signal valid transferred from the exterior of the crossbar routing switch 100.

Detailed Description Text - DETX (29):

FIG. 5 shows external interface signals of the adaptive routing controller corresponding to the adaptive routing controller unit 106 in the **crossbar** core module 101 of FIG. 1. The **arbitration** request unit 104a to 104j in the **crossbar** core module 101 of FIG. 1 interpreted the packet tag of FIG. 2 and drives the normal **arbitration** request signal ReqArb[9 . . . 0] or emergency **arbitration** request signal ReqEmg[9 . . . 0] to the corresponding normal routing controller units 105a to 105h and the corresponding adaptive routing controller unit 106. The input controller modules 102a to 102j of FIG. 1 drive the tail signal tail[9 . . . 0] notifying the rear portion of packet to the corresponding normal routing controller units 105a to 105h and the corresponding adaptive routing controller unit 106 by using the packet valid signal(valid) transferred from the outside of the **crossbar** routing **switch** 100. The output controller modules 103a to 103j of FIG. 1 latch the ready signal(Ready) transferred from the outside of the **crossbar** routing **switch** 100 and drive the ready signal Xready to the corresponding normal routing controller units 105a to 105h and the corresponding adaptive routing controller unit 106.

Detailed Description Text - DETX (40):

As described above, the present invention can implement a **crossbar** routing **switch** of high performance and high function by providing a routing controller adapted to the inherent function and characteristics of the **crossbar** routing **switch** by inventing a priority based round robin **arbitration** method for offering fairness and preventing starvation and a normal routing controller for performing the method and by inventing an adaptive routing controller for performing an adaptive routing control by adding an adaptive routing **switch** logic to two normal routing controllers.

Current US Original Classification - CCOR (1):

710/317

US-PAT-NO: 6247100

DOCUMENT-IDENTIFIER: US 6247100 B1

TITLE: Method and system for transmitting address commands in a multiprocessor system

DATE-ISSUED: June 12, 2001

INVENTOR-INFORMATION: NAME	CITY	STATE	ZIP CODE	COUNTRY
Drehmel; Robert Allen	Goodhue	MN	N/A	N/A
Haselhorst; Kent Harold	Byron	MN	N/A	N/A
Hoover; Russell Dean	Rochester	MN	N/A	N/A
Marcella; James Anthony	Rochester	MN	N/A	N/A

US-CL-CURRENT: 711/141, 709/215 , 710/309 , 710/317 , 711/146

ABSTRACT:

A method and system for transmitting address commands in a multiprocessor system comprising multiple nodes interconnected by an address bus. A request for arbitration of an address bus is transmitted from a controller within a node of multiple nodes to an arbitration switch, which controls transmission across the address bus. The address command is transmitted from the controller to the arbitration switch, in response to receiving a grant of arbitration of the address bus. The address command is then broadcast from the arbitration switch to a controller within each node of multiple nodes, in response to receiving the address command at the arbitration switch. The address command is broadcast from the controller within each node, in response to receiving the broadcast address command at the controller within each node, such that all address command transmissions on the address bus are transmitted to each processor within a multiprocessor system.

8 Claims, 3 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 3

---------- KWIC ---------

Detailed Description Text - DETX (4):

Processor nodes 10a, 10b, 10c, and 10d, main memory subsystems 16a-16n and I/O buses 18a-18n are all connected, preferably via electronically isolated busses, to a non-blocking **crossbar switch** system 14 that provides a communication path between the devices. Non-blocking **crossbar switch** system 14 may include multiple levels of non-blocking **crossbar switches, which arbitrate** between multiple operation requests between devices.

Current US Cross Reference Classification - CCXR (3):

710/317

US-PAT-NO: 6230229

DOCUMENT-IDENTIFIER: US 6230229 B1

TITLE: Method and system for arbitrating path contention in a crossbar interconnect network

DATE-ISSUED: May 8, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Van Krevelen; Christopher J.	Coon Rapids	MN	N/A	N/A
Nelson; Reed S.	Shoreview	MN	N/A	N/A
Hodapp, Jr.; Don J.	Maple Grove	MN	N/A	N/A
Hamre; John D.	Plymouth	MN	N/A	N/A

US-CL-CURRENT: 710/317, 710/107 , 710/113 , 710/241 , 710/38 , 710/39

ABSTRACT:

A method and system for transmitting data among a plurality of cards in a crossbar interconnect network having a plurality of cards each having source paths and destination paths utilizes a plurality of source arbitrators and a plurality of destination arbitrators each associated with the cards. The source arbitrators generate connection request commands from the source paths requesting access to a desired destination path and broadcasts the request for receipt by all of the destination arbitrators. The destination arbitrator associated with the desired destination path captures the connection request command and processes the command based on whether or not the desired destination path is busy. If the desired destination path is not busy, the destination arbitrator generates a connection command requesting a connection be made between the source path and the desired destination path. If the desired destination path is busy, the destination arbitrator stores the connection request command in one of a plurality of buffers until the desired destination path becomes available.

36 Claims, 3 Drawing figures

Exemplary Claim Number: 24

Number of Drawing Sheets: 3

---------- KWIC ---------

Brief Summary Text - BSTX (2):

This invention relates to methods and systems for **arbitrating** path contention in an interconnect fabric utilizing serial or parallel **crossbar switch** technology.

Brief Summary Text - BSTX (5):

In an interconnect fabric using serial or parallel **crossbar switch** technology, path **arbitration** is required when path contention occurs. For example, in a system having an NxN fully connected non-blocking **crossbar,** all of the cards in the system have their input paths fully connected to the **crossbar**. A problem arises when more than one source path needs to send data

to the same destination path at the same time. A source path that is connected to a destination path may remain connected for a long period of time blocking all traffic intended for that path regardless of priority. The source path(s) end up continuously trying to gain access to the destination path until it becomes available without any assurance it will ever get connected to the desired destination path. This is referred to as a lock-out condition when one source path cannot make its connection. A fair system of **arbitration** would capture the connection requests and honor them in the order in which they were received.

Current US Original Classification - CCOR (1):
710/317

US005239629A

United States Patent [19]

Miller et al.

[11] Patent Number: 5,239,629

[45] Date of Patent: Aug. 24, 1993

[54] **DEDICATED CENTRALIZED SIGNALING MECHANISM FOR SELECTIVELY SIGNALING DEVICES IN A MULTIPROCESSOR SYSTEM**

[75] Inventors: **Edward C. Miller; George A. Spix; Anthony R. Schooler,** all of Eau Claire; **Douglas R. Beard,** Eleva; **Alexander A. Silbey; Andrew E. Phelps,** both of Eau Claire, all of Wis.

[73] Assignee: **Supercomputer Systems Limited Partnership,** Eau Claire, Wis.

[21] Appl. No.: **536,192**

[22] Filed: **Jun. 11, 1990**

Related U.S. Application Data

[63] Continuation-in-part of Ser. No. 459,083, Dec. 29, 1989.

[51] Int. Cl.[5] G06F 13/00; G06F 15/16

[52] U.S. Cl. **395/325**; 364/229; 364/242.6; 364/238; 364/937.01

[58] Field of Search ... 364/200 MS File, 900 MS File; 395/200, 325, 425, 800

[56] **References Cited**

U.S. PATENT DOCUMENTS

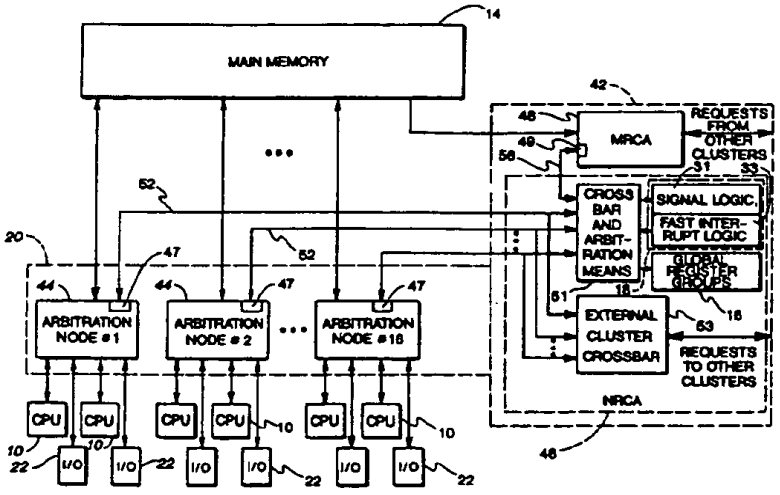
3,510,844	5/1970	Aranyi	364/200
3,676,861	7/1972	Ruth	364/200
3,755,785	8/1973	Kirk	364/200
3,772,656	11/1973	Serracchioli	364/200
3,925,766	12/1975	Bardotti	364/200
3,996,564	12/1976	Kerrigan	364/200
4,034,346	7/1977	Hostein	364/200
4,124,889	11/1978	Kaufman	364/200
4,200,930	4/1980	Rawlings	364/200
4,264,954	4/1981	Briggs	364/200
4,328,543	5/1982	Brickman	364/200
4,418,382	11/1983	Larson	364/200
4,428,043	1/1984	Gatiller	364/200
4,484,270	11/1984	Quernemoen	364/200
4,543,630	9/1985	Neches	395/200
4,636,942	1/1987	Chen et al.	395/725
4,718,006	1/1988	Nishida	395/425
4,745,545	5/1988	Schiffleger	395/325
4,751,634	6/1988	Burrus	364/200
4,754,398	6/1988	Pribnow	395/200
4,807,116	2/1989	Katzman et al.	395/200
4,816,990	3/1989	Williams	395/650
4,845,722	7/1989	Kent et al.	370/58.2
4,891,751	1/1990	Call et al.	345/800
4,905,145	2/1990	Sauber	395/425
4,920,485	4/1990	Vahidsafa	395/725
4,937,733	6/1990	Gillett, Jr. et al.	395/325
5,016,162	5/1991	Epstein et al.	395/775
5,016,167	5/1991	Nguyen et al.	395/725

Primary Examiner—David Y. Eng
Attorney, Agent, or Firm—Patterson & Keough

[57] **ABSTRACT**

A signaling mechanism for sending and receiving signals to and from any one of all of a plurality of devices, including peripheral controllers and processors, in a multiprocessor system. The signaling mechanism includes two switches, a first switch routing a signal command generated by the device to a signal dispatch logic and a second switch for receiving signals generated by the signal dispatch logic and routing the signals to the selected device. The signal dispatch logic receiving the signal command, decodes the destination select value and generates a signal to be sent to the selected device. The signal command includes a destination select value representing a device selectably determined by the device. The signaling mechanism also includes an arbitration mechanism connected to the signal dispatch logic and the first switch for resolving simultaneous conflicting signal commands issued by two or more devices. The signal generated by the signal dispatch logic may include a plurality of bits representing one or more types of predefined signals to be acted upon by the device.

10 Claims, 14 Drawing Sheets



US-PAT-NO: 6125429

DOCUMENT-IDENTIFIER: US 6125429 A

TITLE: Cache memory exchange optimized memory organization for a computer system

DATE-ISSUED: September 26, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Goodwin; Paul M.	Littleton	MA	N/A	N/A
Van Doren; Stephen	Northborough	MA	N/A	N/A

US-CL-CURRENT: 711/143, 710/317 , 711/121 , 711/133 , 711/148 , 711/153

ABSTRACT:

Data coherency in a multiprocessor system is improved and data latency minimized through the use of data mapping "fill" requests from any one of the multiprocessor CPUs such that the information requested is acquired through the crossbar switch from the same memory module to which the "victim" data in that CPUs cache must be rewritten. With such an arrangement rewrite latency periods for victim data within the crossbar switch is minimized and the `ships crossing in the night` problem is avoided.

7 Claims, 6 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 6

---------- KWIC ---------

Detailed Description Text - DETX (2):

Referring now to FIG. 1, a computer system 10 is shown having a **crossbar switch** 12 connected to an **arbiter** chip 14, an input-output device 16, a number of central processing units (CPUs), in this illustrative embodiment numbered 20, 22, 24, 26, and a number of memory modules, numbered 30, 32, 34 and 36. Each of the CPUs, 20-26 will have what is known as a "cache" memory, 40, 42, 44 and 46 respectively, and cache controllers 41, 43, 45 and 47 respectively. Each of the CPUs, 20-26 will take data from a selected one of the four shown memory modules, 30-36.

Detailed Description Text - DETX (4):

With a **crossbar switch** such as 12, each of the four CPUs 20-26 could be connected to one of the four memory modules 30-36, thereby connecting all four CPUs to a separate one of the four memory modules 30-36. A **crossbar** type system may have all four CPUs 20-26 reading data, and all four memory modules 30-36 may be providing data at the same time, versus only one at a time with a bus type system. Thus a **crossbar switch** provides parallelism to the exchange of memory data between the main memory and the cache memories in the CPUs, thereby increasing the throughput of the computing system. If two different CPUs wanted to access data that happened to be in the same memory module, **arbiter** chip 14 would decide which of the CPUs could have access to that memory at that particular time. The nonselected CPU would have to wait until the first CPU finished, for its memory connection and "fill" of data to occur.

Detailed Description Text - DETX (19):

Since the interleave and index portions of the memory are made to be the same between the "victim" and the "fill", then only a single resource is allocated by the **crossbar switch and the arbiter** chip 14. The "victim" data has an assured direct line to the necessary memory resource and less time is consumed in the rewrite operation. Thus use of the present invention assures data coherency in a multi-CPU system and minimizes the latency of the rewrite command for "victim" data.

Current US Cross Reference Classification - CCXR (1):

710/317

US-PAT-NO: 6038630

DOCUMENT-IDENTIFIER: US 6038630 A

TITLE: Shared access control device for integrated system with multiple functional units accessing external structures over multiple data buses

DATE-ISSUED: March 14, 2000

INVENTOR-INFORMATION: NAME	CITY	STATE	ZIP CODE	COUNTRY
Foster; Eric M.	Owego	NY	N/A	N/A
Franklin; Dennis E.	Endicott	NY	N/A	N/A
Jackowski; Stefan P.	Endicott	NY	N/A	N/A
Wallach; David	Raleigh	NC	N/A	N/A

US-CL-CURRENT: 710/317, 709/231 , 710/21 , 710/22 , 710/23 , 710/52 , 711/100 , 711/147 , 711/150 , 711/211

ABSTRACT:

A multi-path access control device for an integrated system is presented which allows simultaneous access to multiple external devices coupled thereto by multiple functional units. The multiple functional units are coupled to the shared access control device across two or more high speed, shared data buses. The control device includes multiple bus ports, each coupled to a different data bus, and a non-blocking crossbar switch coupled to the bus ports for controlling forwarding, with zero cycle latency, of requests from the functional units. Multiple external device ports are coupled to the non-blocking crossbar switch for receiving requests forwarded by the crossbar switch, and each external device is coupled to a different external device port. The crossbar switch allows multiple requests at the bus ports directed to different external devices to be forwarded to different external device ports for simultaneous accessing of different external devices coupled thereto pursuant to the multiple requests.

22 Claims, 3 Drawing figures

Exemplary Claim Number: 10

Number of Drawing Sheets: 3

---------- KWIC ---------

Detailed Description Text - DETX (15):

FIG. 3 depicts one embodiment of a dual path memory controller 230 in accordance with the present invention. In this embodiment, two bus ports 207 & 209 are at one side of the control device, and two external device ports 221 & 223 are at the other. Significant to the control device is a non-blocking **crossbar switch** 240 which receives requests from the dual bus ports (through an **arbitrator** 235) and forwards the requests as appropriate to the external device ports 221 & 223 across respective memory controllers 250 & 260 coupled between the **crossbar switch** and the memory ports.

Detailed Description Text - DETX (16):

Note the data movement on an internal data bus is handled through master/slave communication. A master will request a read or write cycle (or cycles) from a slave device. In the case of multiple masters, there is also an **arbitration** scheme on the bus itself to determine which master function has priority when there are concurrent requests. When applied to the present invention, a master on one of the internal data buses gains control of the bus and requests a read or a write cycle from the slave interface in the **crossbar switch**. Within the **crossbar,** this in translated to the master interface of the correct output port, which in turn passes the request to the slave interface of the associated memory controller (250, 260).

Claims Text - CLTX (10):

7. The multi-path access control device of claim 1, wherein said non-blocking **crossbar switch further comprises an arbitrator for arbitrating** access to said different external devices by said multiple functional units including by said at least two functional units coupled to said at least one data bus of said multiple data buses.

Claims Text - CLTX (38):

19. The integrated system of claim 14, wherein said non-blocking **crossbar switch further comprises an arbitrator for arbitrating** access to said different external devices by said multiple functional units including by said at least two functional units coupled to said at least one data bus of said multiple data buses.

Current US Original Classification - CCOR (1):

710/317

US-PAT-NO: 5896516

DOCUMENT-IDENTIFIER: US 5896516 A

TITLE: Method and apparatus for reducing propagation latency in a high speed crossbar switch

DATE-ISSUED: April 20, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Powell, Jr.; Lawrence Joseph	Round Rock	TX	N/A	N/A
Venkatramani; Krishnamurthy	Austin	TX	N/A	N/A

US-CL-CURRENT: 710/317

ABSTRACT:

A protocol and apparatus for **crossbar switches** where the cycle time is too short to allow the updating of the input/output buffer controls by the **arbitration** logic in one cycle. The **crossbar switch** has separate data paths and commands paths. Two types of commands are sent over the **crossbar switches**. The first type is an address (A) only command which consist of a single packet needing one clock cycle. The second type of command is an Address with Data command (AD), consisting of two through nine packets, and requiring a maximum of nine clock cycles. A command becomes a request through two different paths through the **crossbar switch**. The first path is via an input bypass path which allows an input command buffer to be bypassed and a request written directly to a multiplexer. The second path is through the input command buffer which is written but not selected until processing is completed for the previous command. The **crossbar** protocol allows a request to be accepted by writing information into the output buffers before the accept is available.

13 Claims, 6 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 5

---------- KWIC ---------

Abstract Text - ABTX (1):

A protocol and apparatus for **crossbar switches** where the cycle time is too short to allow the updating of the input/output buffer controls by the **arbitration** logic in one cycle. The **crossbar switch** has separate data paths and commands paths. Two types of commands are sent over the **crossbar switches**. The first type is an address (A) only command which consist of a single packet needing one clock cycle. The second type of command is an Address with Data command (AD), consisting of two through nine packets, and requiring a maximum of nine clock cycles. A command becomes a request through two different paths through the **crossbar switch**. The first path is via an input bypass path which allows an input command buffer to be bypassed and a request written directly to a multiplexer. The second path is through the input command buffer which is written but not selected until processing is completed for the previous command. The **crossbar** protocol allows a request to be accepted by writing information into the output buffers before the accept is available.

Brief Summary Text - BSTX (8):

It is desirable to have a **crossbar switch** protocol for reducing the number of cycles required for updating input/output data buffers by **arbitration** logic in a **crossbar switch** to one cycle.

Brief Summary Text - BSTX (10):

This invention relates to a protocol and apparatus for **crossbar switches** where the cycle time is too short to allow the updating of the input/output buffer controls by the **arbitration** logic in one cycle. The invention is directed to reducing the latency resulting when a request must be delivered, and an accept acted on in the same cycle, which is on the order of 5 nanoseconds. Data from an input port must be propagated to an output port over long wires. The **crossbar switch** has separate data paths and commands paths. Two types of commands are sent over the **crossbar switches**. The first type is an address (A) only command which consists of a single packet needing one clock cycle. The second type of command is an Address with Data (AD) command, consisting of two through nine packets, and requiring up to a maximum of nine clock cycles. However, the first cycle in every command, which is the address cycle, is used for **arbitration**. If the AD command, for example, is accepted, the data will follow and an **arbiter** will ignore following AD commands until the data transfer has completed. A command becomes a request through two different paths through the **crossbar switch**. The first path is via an input bypass path which allows an input command buffer to be bypassed and a request written directly to a request multiplexer. The second path is through the input command buffer which is written, but not selected until processing is completed for the previous command. The **crossbar switch** protocol allows a request to be accepted by writing information into output buffers before the accept is available. If an accept is not generated, a reset line is brought active when the size of an Address/Data request is too large. The **crossbar switch** allows continuous writing of the output buffers every cycle, if an active request is generated from a different port every cycle. If the accept is not activated, the request multiplexer selects a temporary buffer to send the same request again. The **crossbar switch** provides the capability of choosing between resending the same command or sending a command out of the input command buffer.

Current US Original Classification - CCOR (1):

710/317

US-PAT-NO: 5889969

DOCUMENT-IDENTIFIER: US 5889969 A

TITLE: Logical bus structure including plural physical busses for a multiprocessor system with a multi-level cache memory structure

DATE-ISSUED: March 30, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Getzlaff; Klaus Jorg	Schoenaich	N/A	N/A	DE
Leppla; Bernd	Chiningen	N/A	N/A	DE
Tast; Hans-Warner	Weil i. Schoenbuch	N/A	N/A	DE
Wille; Udo	Holzgerlingen	N/A	N/A	DE

US-CL-CURRENT: 710/113, 710/241 , 710/316

ABSTRACT:

An improved multiple bus system for a multiprocessor computer system is disclosed for a computer system having a multiple level cache memory structure. The system includes one or more logical busses each including two or more physical busses for coupling multiple processors to a memory unit. Each logical bus is coupled to a bus switching unit which in turn couples all of the processors in the multiprocessor system to a memory unit over the physical busses comprising the logical bus. The system further manages near end signal reception problems caused by multiple processors electrically interconnected over such a bus system.

12 Claims, 11 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 8

---------- KWIC ---------

Brief Summary Text - BSTX (12):

Such a central **arbitration** concept is disclosed in IBM Technical Disclosure Bulletin, vol. 35, no. 6, November 1992 (FIG. 2) employing a central **crossbar switch**. Herein, memory requests are transferred via a processor bus, and data via the **crossbar switch** to a memory unit. This dividing up concept allows a central memory control for an interleaving access to a number of memory banks.

Current US Cross Reference Classification - CCXR (2):

710/316

US-PAT-NO: 5838937

DOCUMENT-IDENTIFIER: US 5838937 A

TITLE: Data transmitting/receiving method using distributed path control in data switching system

DATE-ISSUED: November 17, 1998

INVENTOR-INFORMATION: NAME	CITY	STATE	ZIP CODE	COUNTRY
Lee; Hyeun Tae	Daejeon	N/A	N/A	KR
Lee; Keun Woo	Daejeon	N/A	N/A	KR

US-CL-CURRENT: **710/316**, 370/351 , 370/360

ABSTRACT:

A data transmitting/receiving method using distributed path control in a data switching system is disclosed, including the steps of: (1) storing data by distributing a first code and a last code of a received data frame, and then reading and storing a frame header; requesting a path setting and then initializing a trial number; transmitting a setup request signal in the next time slot with a destination port address if a polling address is its own address, and thus transmitting the stored header through a data channel when the response signal received from the destination output port indicates an end of the path controlling operation; and transmitting data stored in an input buffer to the end of a frame and requesting a path release at its polling address; (2) releasing the path in case of path release request when the destination port address is its own port address; if the output port is not in use, the output buffer is not is a full state, and a communication channel is available, indicating the end of the path controlling operation in a response bus in case of the path setup request and receiving a switch path setup data to discriminate an initial code and last code of the frame, thereby transmitting the data.

4 Claims, 10 Drawing figures

Exemplary Claim Number: 4

Number of Drawing Sheets: 10

---------- KWIC ---------

Brief Summary Text - BSTX (4):

The internal non-blocking **switch** guarantees that there will be no collision in the path of the **switch** and thus provides an independent path between the input and output. Therefore, as long as the path is set up, a full bandwidth inside the medium is guaranteed. For a prominent example of the non-blocking, there is a **crossbar switch**. However, if a plurality of input ports are requested to be connected to one output port, collisions occur in the output port because the simultaneous connection in pairs is not available. Thus, a means is needed to **arbitrate** the connection between a plurality of input ports and one output port.

Brief Summary Text - BSTX (12):

The data **switching** system using a commercial **crossbar switch** has the

following drawback. When controlling a single crossbar switch, the crossbar switch can connect or cut only one crossbar junction point once. Therefore, if the request for the crossbar junction point is simultaneously input from a plurality of arbitration devices, collisions occur in controlling the crossbar junction point even though the mediating function is distributed in an output port, and thus delays the connection.

Detailed Description Text - DETX (8):

The crossbar switching device 200 provides an internal non-blocking data path between the input port and output port. That is, in the crossbar switching device 200, the path whose output port is different from each other has no collision in the switch. But, the data channel 800 having one output port can be connected to only one path at a time, and thus when the simultaneous path setup from a plurality of input ports to the corresponding output port is requested, an output port collision is generated therein. Therefore, an arbitration device is required to solve the output port collision.

Detailed Description Text - DETX (38):

The output port arbitration part 170 examines the destination port address DST-REQ in step 301. The part 170 determines the path controlling mode if the address is its own port address in step 302. If the request is the path release request, the part 170 displays the end of the path controlling operation in the response bus, asserts CXB-STB response signal to the crossbar switch controlling device 300, and then returns to the initial state in step 303.

Detailed Description Text - DETX (40):

If the communication channel is available, the part 170 indicates the end of the path controlling operation to the response bus signal and transmits it to the path setup controlling part 150 of the origination input port. The output port arbitration part 170 asserts the CXB-STB response bus signal to the crossbar switch controlling device 130 in step 308, the transmission controlling part sends the data through the path, and then stores the data in the output buffer in step 309. The frame transmitting part 180 reads the data stored in the output buffer in step 311, and thus determines the data type in step 312. If the data is the initial code of the frame, the part 180 transmits the frame initial code generation command to the medium transmitting part 190 with the data in step 313. If the data is the general data, the part 180 transmits the general data to the medium transmitting part 190. If the part receives the last code of the frame, the part 180 transmits the frame last code generation command to the medium transmitting part 190 in step 315, and completes the operation.

Detailed Description Text - DETX (49):

Here, with the path setup end response, the output port arbitration part 170 asserts the CXB-STB signal to make the crossbar switch control device 300 generate the corresponding crossbar control signal. The MCLK is a common clock of the time slot cycle. MCLKN is a inverse signal of the MCLK. Therefore, the above timing view illustrates an example of using the rising clock of the MCLK and MCLKN.

Detailed Description Text - DETX (54):

In the invention, as the input port is sequentially processed by POL-ID and the corresponding crossbar switch is controlled after a predetermined period of time slot from the path setup and release request, so that the invention needs not an additional arbiter for solving the collision in the crossbar switch controlling device for a commercially crossbar chip which control the crosspoint sequentially.

Current US Original Classification - CCOR (1):
710/316

US-PAT-NO: 5754792

DOCUMENT-IDENTIFIER: US 5754792 A

TITLE: Switch circuit comprised of logically split switches for parallel transfer of messages and a parallel processor system using the same

DATE-ISSUED: May 19, 1998

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Shutoh; Shinichi	Kokubunji	N/A	N/A	JP
Nakagoshi; Junji	Tokyo	N/A	N/A	JP
Hamanaka; Naoki	Tokyo	N/A	N/A	JP
Takeuchi; Shigeo	Hanno	N/A	N/A	JP
Tanaka; Teruo	Hachioji	N/A	N/A	JP

US-CL-CURRENT: 709/243, 709/249 , 710/317 , 712/11

ABSTRACT:

A parallel processor system including a plurality of processors. When packets of same destination PE number are inputted from different ports, the destination PE number is added with ID numbers of leading ports of split crossbar switches to which the different input ports belong, respectively, by using respective addition circuits, to thereby determine a transfer destination output port for the packets. A plurality of the split crossbar switches having different numbers of input/output ports are realized by partitioning a crossbar switch. By means of an input port select circuit provided in association with each of the output ports, an output request for the packet from the input port belonging to the split crossbar switch to which the associated output port belongs is accepted, while output requests for the packets from the input ports belonging to the other split crossbar switches are inhibited from being accepted, whereby transfer of broadcast packets are inhibited between the split crossbar switches belonging to a physically same crossbar switch. Such situation can be evaded in which same broadcast packets arrive at one and the same processor a number of times.

31 Claims, 19 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 14

---------- KWIC ---------

Detailed Description Text - DETX (11):

FIG. 1 shows a structure of the 9-input/9-output port **crossbar switch** 401 and connection between the **crossbar switch** 401 and a SVP 101. The **crossbar switch** 401 includes for each input/output port a RE (Read Enable) control circuit 103 for reading packets from the input port, a selector 107 and an **arbitration** circuit 104 for controlling the selector 107 to thereby **arbitrate** or manage a plurality of packet transfer requests issued for a same output port. An aspect featuring the present invention is seen in that the addition

circuit 105 and the input port selecting circuit 106 is provided for each of the input/output ports.

Detailed Description Text - DETX (25):
As the one-to-one packet transfer, there can be conceived a packet transfer between a port X00 to which an EX 00 is connected and a port X10 to which the EX 10 is connected, i.e., the packet transfer through the X-crossbar switch, and a packet transfer between a port Y10 to which the EX 10 is connected and a port Y14 to which the EX 14 is connected, i.e., the packet transfer through the Y-crossbar switch, as shown in FIG. 3. Now, let's consider the packet transfer from the EX 00 to the EX 10 by reference to FIG. 1. In the following description, emphasis will be put on the RE control circuit 103-0, an addition circuit 105-0 having the input port number "0", the input port select circuit 106-1 and the arbitration circuit 104-1 of the output port number "1".

Detailed Description Text - DETX (35):
In conjunction with the packet transfer through the crossbar switch 401-0, there are available two transfers, i.e., a transfer from the port X00 to which the EX 00 is connected to the ports X00 to X10 and a transfer from the port Y10 to which the EX 10 is connected to the ports Y10 to Y14. The following description will be directed to the transfer from the EX 00 to the EXs 00 to 10 with reference to FIG. 1, by paying attention primarily on the RE control circuit 103-0 and the addition circuit of the input port number "0" as well as the input port select circuits 106-0 and 106-1 and the arbitration circuits 104-0 and 104-1.

Current US Cross Reference Classification - CCXR (2):
710/317

US-PAT-NO: 5239629

DOCUMENT-IDENTIFIER: US 5239629 A

TITLE: Dedicated centralized signaling mechanism for selectively signaling devices in a multiprocessor system

DATE-ISSUED: August 24, 1993

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Miller; Edward C.	Eau Claire	WI	N/A	N/A
Spix; George A.	Eau Claire	WI	N/A	N/A
Schooler; Anthony R.	Eau Claire	WI	N/A	N/A
Beard; Douglas R.	Eleva	WI	N/A	N/A
Silbey; Alexander A.	Eau Claire	WI	N/A	N/A
Phelps; Andrew E.	Eau Claire	WI	N/A	N/A

US-CL-CURRENT: 710/317

ABSTRACT:

A signaling mechanism for sending and receiving signals to and from any one of all of a plurality of devices, including peripheral controllers and processors, in a multiprocessor system. The signaling mechanism includes two switches, a first switch routing a signal command generated by the device to a signal dispatch logic and a second switch for receiving signals generated by the signal dispatch logic and routing the signals to the selected device. The signal dispatch logic receiving the signal command, decodes the destination select value and generates a signal to be sent to the selected device. The signal command includes a destination select value representing a device selectably determined by the device. The signaling mechanism also includes an arbitration mechanism connected to the signal dispatch logic and the first switch for resolving simultaneous conflicting signal commands issued by two or more devices. The signal generated by the signal dispatch logic may include a plurality of bits representing one or more types of predefined signals to be acted upon by the device.

10 Claims, 17 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 14

---------- KWIC ---------

Detailed Description Text - DETX (5):

Referring now to FIG. 9, the cross bar/**arbitration** means 51 is described in greater detail. The flow begins with data from one of the **arbitration** nodes 44 which has been buffered by the NRCA means 46. As each request is received at the NRCA input registers 510 (FIG. 10), decode logic 406 decodes the request to be presented to a global register **arbitration** network 410. If simultaneous requests come in for multiple global registers 16 in the same global register file 400, these requests are handled in a pipelined manner by the FIFO's 412,

pipelines 414 and the global register arbitration network 410. Priority is assigned by a FIFO (first in, first out) scheme supplemented with a multiple request toggling priority scheme. The global register arbitration network 410 uses this type of arbitration logic, or its equivalent, to prioritize simultaneous requests to the same global register file 400. When priority is determined by the arbitration network 410, a 17.times.10 crossbar switch means 430 matches the request in the FIFO 412 with the appropriate global register file 400. A plurality of NRCA input registers 510 (FIG. 10) provide seventeen paths into the global registers input crossbar 430. There are eight paths 440 out of the global registers input crossbar 430 to the global register files 400, one path 442 to the signal logic 31, and one path 444 to the fast interrupt logic 33. After the global register file operation is completed, global register output cross bar 422 routes any output from the operation back to the requesting port.

Current US Original Classification - CCOR (1):
710/317

US-PAT-NO: 5179669

DOCUMENT-IDENTIFIER: US 5179669 A

TITLE: Multiprocessor interconnection and access arbitration arrangement

DATE-ISSUED: January 12, 1993

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Peters; Daniel V.	Warrenville	IL	N/A	N/A

US-CL-CURRENT: 710/317, 370/310 , 370/462

ABSTRACT:

In a multiprocessor system (FIG. 1), the processors (10-12) are interconnected by a non-blocking communication medium such as a crossbar switch (19). Each processor is connected to a dedicated port circuit (18) at the switch by an optical link (16). Each port circuit is connected to the crossbar switch by an electrical link (20). The port circuits are interconnected by a contention medium (14). A port circuit sends an access request by its connected processor to the destination processor over the contention medium. Circuitry (205) at each port circuit receives requests, for access to the connected processor, prioritizes conflicting requests, and grants them sequentially. The circuitry interleaves grants of access to the connected processor with grants of outgoing access requests made by the connected processor. The circuitry grants an access request by causing the crossbar switch to establish the corresponding connection.

19 Claims, 8 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 8

---------- KWIC ---------

Brief Summary Text - BSTX (6):

Because a non-blocking interconnection medium such as a **crossbar switch** theoretically allows any desired connection between any processors to be made at any time, **arbitration** of access to the medium itself is not necessary. This situation is unlike typical bus **arbitration**. In single bus **arbitration,** the bus is the resource in demand and the destination unit, e.g., processor, is assumed by the **arbiter** to be available. Conversely, in the situation of a non-blocking interconnection medium, the medium is assumed to be available and the destination unit, or the link(s) connecting the destination unit to the medium, is the resource in demand.

Brief Summary Text - BSTX (8):

Many different **arbitration** arrangements are known in the art. For example, centralized **arbitration** in a central control unit has been practiced in conjunction with **crossbar switches** in computer and telephone systems. Although centralized controller **arbitration** arrangements operate suitably to perform their intended function, they are not always desirable, because of the inherent system complexity resulting from the many interconnections required between the

controller, the interconnection medium, and the interconnected units. Also, the centralized arrangements tend to be slower and more complex than other arrangements, and they typically do not allow for modular growth of the system. Furthermore, a reliability problem exists with such arrangements, since a malfunction of the controller may remove the whole system from operation. While it is true that this fault intolerance may be overcome by means of replicating the central control unit, it is an expensive and complex proposition.

Brief Summary Text - BSTX (9):

It is known to use distributed **arbitration** arrangements, in which a central controller is not used to determine access and instead the interaction of the requesting units determines access in the event of simultaneous requests. Such distributed arrangements are often preferable, since the expense, complexity, and slowness of, and the reliability problems associated with, the centralized controller arrangements are avoided while modular growth of the system is facilitated. But an efficient arrangement of this nature that could be applied to a nonblocking medium such as a **crossbar switch** and provide fair **arbitration,** i.e., **arbitration** which does not favor certain processors in their access requests while "starving out" other processors from requested access, and one which eases bottlenecks and also eliminates the possibility of deadlock between processors, has not been available.

Brief Summary Text - BSTX (11):

This invention is directed to solving these and other disadvantages of the prior art. According to the invention, a multiprocessor system which comprises a plurality of processors interconnected by a non-blocking communication medium, such as a **crossbar switch,** further includes a unique inter-processor access contention arrangement that enables each processor to independently **arbitrate** incoming access requests from other processors to itself, and illustratively also to **arbitrate** the incoming requests with its own outgoing access requests. The arrangement is structured as follows. A contention communication medium interconnects the processors. Each processor includes a facility for communicating a request to access another processor to the other processor across the contention medium. Each processor further includes a facility which responds to conflicting requests for access to its own processor that it receives from other processors across the contention medium, by resolving the conflicts and granting the requests to access its own processor sequentially. The arrangement further includes a facility that responds to the granting of an access request by establishing a connection through the non-blocking medium between the requesting processor whose request for access has been granted and the granting processor.

Current US Original Classification - CCOR (1):

710/317

US-PAT-NO: 5053942

DOCUMENT-IDENTIFIER: US 5053942 A
See image for Certificate of Correction

TITLE: Bit-sliced cross-connect chip having a tree topology of arbitration cells for connecting memory modules to processors in a multiprocessor system

DATE-ISSUED: October 1, 1991

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Srini; Vason P.	Berkeley	CA	N/A	N/A

US-CL-CURRENT: 710/317

ABSTRACT:

A cross-connect circuit for coupling each of a plurality of processors to a memory module selected from a plurality of such modules, provided the module in question has not been identified for connection to another of the processors is disclosed. The circuit is preferably organized as a bit-sliced chip. The connections made by the cross-connect circuit can be changed after each memory cycle.

7 Claims, 7 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 7

---------- KWIC ---------

Detailed Description Text - DETX (11):
The output of each **arbiter** controls a set of **switches in a crosspoint** matrix 20. **Crosspoint** matrix 20 makes the actual connections between each processor 12 and the memory module 22 requested thereby. The connections in question are made on three lines shown at 32. One of these lines is used to communicate one bit of an address specifying a data word in module 22. The other two lines are used to communicate one bit of the data word stored at the address in question, one for reading the bit from the memory module and one for writing the bit thereto.

Detailed Description Text - DETX (30):
As noted above, once priority has been assigned by the **arbiter, the crosspoint** matrix on the chip makes the actual connections. The **crosspoint** matrix connects the 16 processors to the 32 memory modules using 512 "**switches**". Each **switch** connects the three lines from the processor to the corresponding three lines of the memory module in question. The first of these three lines transmits one bit of the requested address in the memory module. The second line is used to transmit a bit of a data word to be stored at the address in question. And, the third line is used to receive one bit of the data word stored at the address in question if a read operation is being performed on the memory module in question

Detailed Description Text - DETX (31):

The preferred embodiment of the **crosspoint** matrix is shown in FIG. 6 at 500. **Crosspoint** matrix 500 is organized as 32 rows 502 and 16 columns 504. Each row 502 corresponds to a memory module and consists of the three conductors discussed above. Each column 504 corresponds to a processor and consists of the corresponding three conductors. Each **switch** 508 is connected to a particular row and column and consists of a combinational logic block which connects the three conductors of the row and column in question when the **switch** is activated. The Ith **switch** in the Jth column is controlled by a signal G.sub.I-J which is generated by one of the **arbiters** discussed above. G.sub.I-J is the Ith priority granting signal from the **arbiter** corresponding the Jth memory module.

Current US Original Classification - CCOR (1):
710/317

US-PAT-NO: 6038630

DOCUMENT-IDENTIFIER: US 6038630 A

TITLE: Shared access control device for integrated system with multiple functional units accessing external structures over multiple data buses

DATE-ISSUED: March 14, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Foster; Eric M.	Owego	NY	N/A	N/A
Franklin; Dennis E.	Endicott	NY	N/A	N/A
Jackowski; Stefan P.	Endicott	NY	N/A	N/A
Wallach; David	Raleigh	NC	N/A	N/A

US-CL-CURRENT: **710/317**, 709/231 , 710/21 , 710/22 , 710/23 , 710/52 , 711/100 , 711/147 , 711/150 , 711/211

ABSTRACT:

A multi-path access control device for an integrated system is presented which allows **simultaneous access** to multiple external devices coupled thereto by multiple functional units. The multiple functional units are coupled to the shared access control device across two or more high speed, shared data buses. The control device includes multiple bus ports, each coupled to a different data bus, and a non-blocking crossbar switch coupled to the bus ports for controlling forwarding, with zero cycle latency, of requests from the functional units. Multiple external device ports are coupled to the non-blocking crossbar switch for receiving requests forwarded by the crossbar switch, and each external device is coupled to a different external device port. The crossbar switch allows multiple requests at the bus ports directed to different external devices to be forwarded to different external device ports for **simultaneous accessing** of different external devices coupled thereto pursuant to the multiple requests.

22 Claims, 3 Drawing figures

Exemplary Claim Number: 10

Number of Drawing Sheets: 3

---------- KWIC ---------

Abstract Text - ABTX (1):

A multi-path access control device for an integrated system is presented which allows **simultaneous access** to multiple external devices coupled thereto by multiple functional units. The multiple functional units are coupled to the shared access control device across two or more high speed, shared data buses. The control device includes multiple bus ports, each coupled to a different data bus, and a non-blocking crossbar switch coupled to the bus ports for controlling forwarding, with zero cycle latency, of requests from the functional units. Multiple external device ports are coupled to the non-blocking crossbar switch for receiving requests forwarded by the crossbar switch, and each external device is coupled to a different external device

port. The crossbar switch allows multiple requests at the bus ports directed to different external devices to be forwarded to different external device ports for **simultaneous accessing** of different external devices coupled thereto pursuant to the multiple requests.

Brief Summary Text - BSTX (9):

Briefly summarized, this invention comprises in one aspect a multi-path access control device which allows **simultaneous accessing** of multiple external devices, to be coupled to the control device, by multiple functional units connected to the control device. The control device includes multiple bus ports coupled to multiple data buses, wherein each bus port receives signals from a different data bus, and the functional units are coupled to the data buses for sending requests thereon to access the multiple external devices. A non-blocking crossbar switch is coupled to the bus ports for receiving and controlling forwarding of requests from the functional units. Multiple external device ports, coupled to the non-blocking crossbar switch, receive requests forwarded by the crossbar switch from the functional units, wherein the non-blocking crossbar switch allows multiple requests at the bus ports to be directed to different external devices simultaneously.

Detailed Description Text - DETX (2):

Generally stated, this invention is targeted to integrated systems having several functions on a single chip wherein the functions access multiple external devices. Pursuant to the invention, a shared, multi-path access control device is employed between multiple, shared internal data buses of the integrated system and multiple ports to the external devices. This control device allows a request on any internal data bus to be sent to any device port. Further, the control device allows multiple requests directed to different external devices to be simultaneously forwarded from multiple internal data buses to the appropriate external device ports for **simultaneous (or same cycle) accessing** of the different external devices. FIG. 1 depicts one embodiment of these general concepts of the present invention.

Detailed Description Text - DETX (15):

FIG. 3 depicts one embodiment of a dual path memory controller 230 in accordance with the present invention. In this embodiment, two bus ports 207 & 209 are at one side of the control device, and two external device ports 221 & 223 are at the other. Significant to the control device is a non-blocking **crossbar switch** 240 which receives requests from the dual bus ports (through an **arbitrator** 235) and forwards the requests as appropriate to the external device ports 221 & 223 across respective memory controllers 250 & 260 coupled between the **crossbar switch** and the memory ports.

Detailed Description Text - DETX (16):

Note the data movement on an internal data bus is handled through master/slave communication. A master will request a read or write cycle (or cycles) from a slave device. In the case of multiple masters, there is also an **arbitration** scheme on the bus itself to determine which master function has priority when there are concurrent requests. When applied to the present invention, a master on one of the internal data buses gains control of the bus and requests a read or a write cycle from the slave interface in the **crossbar switch**. Within the **crossbar,** this in translated to the master interface of the correct output port, which in turn passes the request to the slave interface of the associated memory controller (250, 260).

Claims Text - CLTX (1):

1. A multi-path access control device allowing **simultaneous accessing** of multiple external devices to be coupled thereto by multiple functional units coupled to said multi-path access control device, said multi-path access control device comprising:

Claims Text - CLTX (4):
multiple external device ports coupled to said non-blocking crossbar switch for receiving requests forwarded by said non-blocking crossbar switch from said multiple functional units, wherein multiple requests at said multiple bus ports directed to different external devices can be simultaneously forwarded by said non-blocking crossbar switch through different external device ports of said multiple external device ports for **simultaneous accessing** of said different external devices pursuant to said multiple requests.

Claims Text - CLTX (10):
7. The multi-path access control device of claim 1, wherein said non-blocking **crossbar switch further comprises an arbitrator for arbitrating** access to said different external devices by said multiple functional units including by said at least two functional units coupled to said at least one data bus of said multiple data buses.

Claims Text - CLTX (13):
10. A multi-path access control device allowing **simultaneous accessing** of multiple external devices to be coupled thereto by multiple functional units coupled to said multi-path access control device, said multi-path access control device comprising:

Claims Text - CLTX (16):
multiple external device ports coupled to said non-blocking crossbar switch for receiving requests forwarded by said non-blocking crossbar switch from said multiple functional units, wherein multiple requests at said multiple bus ports directed to different external devices can be simultaneously forwarded by said non-blocking crossbar switch through different external device ports of said multiple external device ports for **simultaneous accessing** of said different external devices pursuant to said multiple requests;

Claims Text - CLTX (21):
12. The multi-path access control device of claim 11, wherein a request on said first data bus to access said SDRAM and a request on said second data bus to access said DRAM can be processed simultaneously by said non-blocking crossbar switch with zero latency so that said request on said first data bus is processed to **access said SDRAM simultaneous** with said request on said second data bus being processed to access said DRAM.

Claims Text - CLTX (22):
13. A multi-path access control device allowing **simultaneous accessing** of multiple external devices to be coupled thereto by multiple functional units coupled to said multi-path access control device, said multi-path access control device comprising:

Claims Text - CLTX (25):
multiple external device ports coupled to said non-blocking crossbar switch for receiving requests forwarded by said non-blocking crossbar switch from said multiple functional units, wherein multiple requests at said multiple bus ports directed to different external devices can be simultaneously forwarded by said non-blocking crossbar switch through different external device ports of said multiple external device ports for **simultaneous accessing** of said different external devices pursuant to said multiple requests; and

Claims Text - CLTX (33):
(iii) multiple external device ports coupled to said non-blocking crossbar switch for receiving requests forwarded by said non-blocking crossbar switch

from said multiple functional units, each external device port to be coupled to a different external device, wherein multiple requests at said multiple bus ports directed to different external devices can be simultaneously forwarded by said non-blocking crossbar switch to different external device ports of said multiple external device ports for **simultaneous accessing** of said different external devices.

Claims Text - CLTX (38):
19. The integrated system of claim 14, wherein said non-blocking **crossbar switch further comprises an arbitrator for arbitrating** access to said different external devices by said multiple functional units including by said at least two functional units coupled to said at least one data bus of said multiple data buses.

Claims Text - CLTX (46):
(iii) multiple external device ports coupled to said non-blocking crossbar switch for receiving requests forwarded by said non-blocking crossbar switch from said multiple functional units, each external device port to be coupled to a different external device, wherein multiple requests at said multiple bus ports directed to different external devices can be simultaneously forwarded by said non-blocking crossbar switch to different external device ports of said multiple external device ports for **simultaneous accessing** of said different external devices; and

Claims Text - CLTX (54):
(iii) multiple external device ports coupled to said non-blocking crossbar switch for receiving requests forwarded by said non-blocking crossbar switch from said multiple functional units, each external device port to be coupled to a different external device, wherein multiple requests at said multiple bus ports directed to different external devices can be simultaneously forwarded by said non-blocking crossbar switch to different external device ports of said multiple external device ports for **simultaneous accessing** of said different external devices;

Current US Original Classification - CCOR (1):
710/317

US-PAT-NO: 6636933

DOCUMENT-IDENTIFIER: US 6636933 B1

TITLE: Data storage system having crossbar switch with multi-staged routing

DATE-ISSUED: October 21, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
MacLellan; Christopher S.	Walpole	MA	N/A	N/A
Walton; John K.	Mendon	MA	N/A	N/A

US-CL-CURRENT: **710/317**, 711/114 , 711/152 , 714/5

ABSTRACT:

A memory system having a backplane with a plurality of receiving slots. Each one of the slots has electrical contacts for providing an indication of Such one of the slots. Each one of the slots has a different slot indication. A plurality of memory boards is provided. Each one of the memory boards is plugged into a corresponding one of the slots. Each one of such boards is coupled to the electrical contacts in the corresponding one of the slots to provide a slot signal indicative of the slot indication provided by the electrical contacts. Each one of such boards has: a memory array region; and a switching network for transferring information between a port of the switching network and a memory on such memory boards The transfer is initiated by a director coupled to such port. The director designates a selected one of the plurality of memory boards. The director provides to the switching network a "tag" indicating such designated one of one of the plurality of memory boards having the memory involved in the requested transfer. The switch network includes a memory board checker, for comparing the slot signal with the "tag" for indicating whether the memory board receiving the "tag" is the director designated one of the plurality of memory boards.

12 Claims, 47 Drawing figures

Exemplary Claim Number: 5

Number of Drawing Sheets: 43

---------- KWIC ---------

Detailed Description Text - DETX (20):

Referring now to FIG. 7, an exemplary one of the director boards 190.sub.1 -190.sub.8, 210.sub.1 -210.sub.8, here director board 190.sub.1 is shown to include directors 180.sub.1, 180.sub.3, 180.sub.5 and 180.sub.7. An exemplary one of the directors 180.sub.1 -180.sub.4, here director 180.sub.1 is shown in detail to include the data pipe 316, the message engine/CPU controller 314, the RAM 312, and the CPU 310 all coupled to the CPU interface bus 317, as shown. The exemplary director 180.sub.1 also includes: a local cache memory 319 (which is coupled to the CPU 310); the **crossbar switch** 318; and, the **crossbar switch** 320, described briefly above in connection with FIGS. 5 and 6. The data pipe 316 includes a protocol translator 400, a quad port RAM 402 and a quad port RAM controller 404 arranged as shown. Briefly, the protocol translator 400 converts between the protocol of the host computer 120, in the case of a front-end director 180.sub.1 -180.sub.32, (and between the protocol used by the

disk drives in bank 140 in the case of a back-end director 200.sub.1 -200.sub.32) and the protocol between the directors 180.sub.1 -180.sub.3, 200.sub.1 -200.sub.32 and the global memory 220 (FIG. 2). More particularly, the protocol used the host computer 120 may, for example, be fiber channel, SCSI, ESCON or FICON, for example, as determined by the manufacture of the host computer 120 while the protocol used internal to the system interface 160 (FIG. 2) may be selected by the manufacturer of the interface 160. The quad port RAM 402 is a FIFO controlled by controller 404 because the rate data coming into the RAM 402 may be different from the rate data leaving the RAM 402. The RAM 402 has four ports, each adapted to handle an 18 bit digital word. Here, the protocol translator 400 produces 36 bit digital words for the system interface 160 (FIG. 2) protocol, one 18 bit portion of the word is coupled to one of a pair of the ports of the quad port RAM 402 and the other 18 bit portion of the word is coupled to the other one of the pair of the ports of the quad port RAM 402. The quad port RAM has a pair of ports 402A, 402B, each one of to ports 402A, 402B being adapted to handle an 18 bit digital word. Each one of the ports 402A, 402B is independently controllable and has independent, but **arbitrated,** access to the memory array within the RAM 402. Data is transferred between the ports 402A, 402B and the cache memory 220 (FIG. 2) through the **crossbar switch** 318, as shown.

Detailed Description Text - DETX (45):

The lower port interface sections W-Z provides address, control, DATA and routing to one of the four of the logic sections 5010.sub.1 -5010.sub.8 (FIGS. 9A, 9B and 9C) in a manner to be described. Each one of the lower interface sections W-Z is adapted to couple a corresponding one of the four memory array regions R.sub.1 -R.sub.4 (FIGS. 9A, 9B and 9C), respectively, via logic sections 5010.sub.1 -5010.sub.8. Each one of the four lower interface sections W-Z independently acts as an arbiter between the four upper interface sections A-D and the logic section 5010.sub.1 -5010.sub.8 coupled thereto. This allows for **simultaneous transfers** (i.e., information cycles) to multiple memory array regions R.sub.1 -R.sub.4 from multiple upper interface sections A-D. The upper interface section A-D are single threaded, i.e., one information cycle must be complete before another information cycle is allowed to the same memory array regions R.sub.1 -R.sub.4.

Detailed Description Text - DETX (47):

More particularly, assume for example that information at upper port 5006.sub.4 (FIGS. 9A, 9B and 9C) of **crossbar switch** 5004.sub.4 is to be transferred to memory array region R.sub.1. Referring to FIG. 10 a negotiation, i.e., **arbitration,** must be made by lower port interface W as a result of a request made by the upper port interface section D of **crossbar switch** 5004.sub.4 to section interface W thereof. When interface section W is available to satisfy such request, (i.e., not satisfying request from other one of the upper port interface sections A-C) interface W issues a grant to upper interface section D.

Current US Original Classification - CCOR (1):

710/317

L Number	Hits	Search Text	DB	Time stamp
1	20	(((switch$3) same (crossbar$1 or crosspoint$1)) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/12/22 16:10
2	3	(simultaneous near3 access$3) and ((((switch$3) same (crossbar$1 or crosspoint$1)) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/12/22 16:13
3	4	(simultaneous$2 near3 transfer$5) and ((((switch$3) same (crossbar$1 or crosspoint$1)) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/12/22 16:13
4	4	((simultaneous$2 near3 transfer$5) and ((((switch$3) same (crossbar$1 or crosspoint$1)) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.)) not ((simultaneous near3 access$3) and ((((switch$3) same (crossbar$1 or crosspoint$1)) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/12/22 16:13

US-PAT-NO: 6138185

DOCUMENT-IDENTIFIER: US 6138185 A

TITLE: High performance crossbar switch

DATE-ISSUED: October 24, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Nelson; Jeffrey J.	Louisville	CO	N/A	N/A
Jessop; Ken N.	Arvada	CO	N/A	N/A

US-CL-CURRENT: 710/33, **710/317** , 710/38

ABSTRACT:

A switch having a plurality of input/output (I/O) ports and a crossbar device programmably coupling a first of the I/O ports with a second of the I/O ports. A plurality of port request controllers (PRCs) are coupled such that each PRC is associated with one of the I/O ports. A plurality of serial request busses are arranged such that each serial request bus couples each PRC with its associated port. A plurality of serial response busses are coupled such that each serial response bus coupling each PRC with its associated PRC. In operation, the serial request and response busses operate independently in a non-blocking fashion to process connection and clear requests in parallel.

24 Claims, 6 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 6

---------- KWIC ---------

Detailed Description Text - DETX (11):

A port is busy if its transmitter is in use and is otherwise available. Busy bit logic 307 includes interface logic for receiving request messages from PRCs 305 and sending reply messages to appropriate PRCs 305. The interface logic of busy bit logic 307 is point-to-point allowing multiple requests to either the same port 201 or different ports 201 to occur simultaneously. When multiple connections occur to different ports the respective PRQs 305 then **arbitrate to the crossbar** connection logic (CCL) 302 to establish a connection.

Detailed Description Text - DETX (12):

Connection setup in the CCL consumes many clock cycles and **arbitration** between ports contending for the same connection resources can add further latency. In a preferred implementation, busy bit logic 307 includes devices to determine if the requesting port was last connected to the destination. Each PRC 201 is aware of whether it was the last port to request connection to the destination port specified in the request (i.e., whether a connection remains set up to the requested destination port). In such a case, the already established **crossbar** connection can be used and a new **crossbar** connection need not be made and efficiency is increased.

Detailed Description Text - DETX (35):

Because the present invention enables multiple concurrent connection requests to be processed, latency associated with multicast and broadcast connection setup is greatly reduced. For any multicast or broadcast group the requests are all performed within the same **arbitration** window with the available member ports being latched for the crossbar connection logic 203.

Current US Cross Reference Classification - CCXR (1):
710/317

L Number	Hits	Search Text	DB	Time stamp
1	20	(((switch$3) same (crossbar$1 or crosspoint$1)) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/12/22 16:16
2	3	(simultaneous near3 access$3) and ((((switch$3) same (crossbar$1 or crosspoint$1)) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/12/22 16:13
3	4	(simultaneous$2 near3 transfer$5) and ((((switch$3) same (crossbar$1 or crosspoint$1)) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/12/22 16:13
4	4	((simultaneous$2 near3 transfer$5) and ((((switch$3) same (crossbar$1 or crosspoint$1)) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.)) not ((simultaneous near3 access$3) and ((((switch$3) same (crossbar$1 or crosspoint$1)) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/12/22 16:13
5	26	((crossbar$1 or crosspoint$1) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/12/22 16:17
6	6	(((crossbar$1 or crosspoint$1) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.) not ((((switch$3) same (crossbar$1 or crosspoint$1)) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/12/22 16:17

US-PAT-NO: 6480941

DOCUMENT-IDENTIFIER: US 6480941 B1

TITLE: Secure partitioning of shared memory based multiprocessor system

DATE-ISSUED: November 12, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Franke; Hubertus	Cortlandt Manor	NY	N/A	N/A
Giampapa; Mark Edwin	Irvington	NY	N/A	N/A
Jann; Joefon	Ossining	NY	N/A	N/A
Joseph; Douglas James	Danbury	CT	N/A	N/A
Pattnaik; Pratap Chandra	Ossining	NY	N/A	N/A

US-CL-CURRENT: 711/153, 710/317 , 711/173

ABSTRACT:

A method and apparatus for sharing memory in a multiprocessor computing system. More specifically, this invention provides a number of system buses with each bus being connected to a respective memory controller which controls a corresponding partition of the memory. Any one of the processors can use any one of the system buses to send real addresses to the connected memory controller which then converts the real addresses into physical addresses corresponding to the partition of memory that is controlled by the receiving memory controller. The processors can be dynamically assigned to different partitions of the memory by via a switching mechanism.

13 Claims, 5 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 6

---------- KWIC ---------

Brief Summary Text - BSTX (19):

Rather than providing a single system bus, several complete internal system buses are provided. Each said internal system bus provides all bus signals including address, data, and control signals. The number of internal system buses determines the number of independent partitions that can be established. The system bus components are part of a cross bar switch. In addition to the system buses, the cross bar switch provides a set of external connector buses. Single system components such as CPUs, I/O controllers, interrupt controllers, but not the memory controllers, can be attached to the external connector bus. Within the **crossbar** switch, the external connector buses can be coupled with the internal system buses. This coupling is setup via a partitioning control unit, also known as the partitioning control unit or coupling control unit. An external connector bus can only be attached to one internal bus at a time. All system components that are attached via the bus coupling of the same internal bus belong to a partition. Memory access in this system is conceptually provided via a multi-port memory controller. A multi-port memory controller

provides several ports to accept memory requests from different sources at a time. Associated with each port is a memory controller interface that implements the proper bus protocol, by snooping on the bus for memory transactions and forwarding them to the memory controller to be realized (e.g. loaded or stored). The memory controller core **arbitrates** among the multiple ports to provide proper service of load and store requests. In the system of this invention, a memory controller interface is attached to each internal system bus. Furthermore, the memory controller interface, in addition to performing the functions described above, implements a real-to-physical remapping. Before memory requests are forwarded to the memory controller core, the real address as provided on the internal system bus is translated into a physical address. The real-to-physical map is setup and maintained by the partitioning control unit. If maps of different partitions overlap, a non-cache coherent inter partition shared memory can be implemented. As described above, the system isolates partitions from each other, provides secure and flexible partitioning through means of configurable "internal system bus to external connector bus coupling". The **crossbar** switches can be connected together along their internal buses to create larger systems.

Detailed Description Text - DETX (7):

In the preferred embodiment we connect an interrupt controller (103) to each internal bus through a connection port (260). In another embodiment, the interrupt controllers are placed inside the crossbar **switch** right onto the internal buses. The latter method eliminates the necessity for accomplished with a line coupling device (310) for each such pair. Each of the k*(n+m) bus coupling devices is connected via two lines, Couple (340) and Decouple (341) to the **switch** control unit (290). The **switch** control unit has to raise either of the lines at system configuration time to either put the attached bus coupling device into an active or into an inactive state. Raising a line for a given period of time will put the bus coupling device into the desired state. The control unit is controlled through a control interface (291). For instance the interface can be either a configuration table (e.g. a truth table, indicating whether IB.sub.i is connected to EC.sub.j) or a programming interface that responds to simple commands such as<connect(ij)>. Interfaces of such kind are well known to those skilled in the art. Once the coupling control unit (290) receives the "instructions" for the connectivity setup via the its interface (291), it utilizes the coupling logic (354) to drive two **multiplexers** (351). One **multiplexer** is used for selecting a specific coupling line (340), and the other one is used for selecting a specific decoupling line (341). Hence, both **multiplexers** (351) must be able to address k*(n+m) different lines connected to the individual bus coupling devices.

Current US Cross Reference Classification - CCXR (1):

710/317

US-PAT-NO: 5081575

DOCUMENT-IDENTIFIER: US 5081575 A

TITLE: Highly parallel computer architecture employing crossbar switch with selectable pipeline delay

DATE-ISSUED: January 14, 1992

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Hiller; John	New York	NY	N/A	N/A
Johnsen; Howard	Granite Spring	NY	N/A	N/A
Mason; John	Ramsey	NJ	N/A	N/A
Mulhearn; Brian	Paterson	NJ	N/A	N/A
Petzinger; John	Oakland	NJ	N/A	N/A
Rosal; Joseph	Bronx	NY	N/A	N/A
Satta; John	White Plains	NY	N/A	N/A
Shurko; Gerald	Ramsey	NJ	N/A	N/A
Solowiejczyk; Yedidiah	New York	NY	N/A	N/A
Stamm; Kenneth	New York	NY	N/A	N/A

US-CL-CURRENT: 710/317

ABSTRACT:

A crossbar **switch** which connects N (N=2.sup.k ; k=0, 1, 2, 3) coarse grain processing elements (rated at 20 million floating point operations per second) to a plurality of memories provides for a parallel processing system free of memory conflicts over a wide range of arithmetic computations (i.e. scalar, vector and matrix). The configuration of the crossbar **switch**, i.e., the connection between each processing element unit and each parallel memory module, may be changed dynamically on a cycle-by-cycle basis in accordance with the requirements of the algorithm under execution. Although there are certain crossbar usage rules which must be obeyed, the data is mapped over parallel memory such that the processing element units can access and operate on input streams of data in a highly parallel fashion with an effective memory transfer rate and computational throughput power comparable in performance to present-day supercomputers. The crossbar **switch** is comprised of two basic sections; a **multiplexer** and a control section. The **multiplexer** provides the actual **switching** of signal paths, i.e. connects each processing element unit to a particular parallel memory on each clock cycle (100 nsec). The control section determines which connections are made on each clock cycle in accordance with the algorithm under execution. Selectable pipelined delay in the control section provides for optimal data transfer efficiency between the processors and memory modules over a wide range of array processing algorithms. The crossbar **switch** also provides for graceful system degradation in computational throughput power without the need to download a new program.

11 Claims, 17 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 17

---------- KWIC ---------

Abstract Text - ABTX (1):

A crossbar **switch** which connects N (N=2.sup.k ; k=0, 1, 2, 3) coarse grain processing elements (rated at 20 million floating point operations per second) to a plurality of memories provides for a parallel processing system free of memory conflicts over a wide range of arithmetic computations (i.e. scalar, vector and matrix). The configuration of the crossbar **switch**, i.e., the connection between each processing element unit and each parallel memory module, may be changed dynamically on a cycle-by-cycle basis in accordance with the requirements of the algorithm under execution. Although there are certain crossbar usage rules which must be obeyed, the data is mapped over parallel memory such that the processing element units can access and operate on input streams of data in a highly parallel fashion with an effective memory transfer rate and computational throughput power comparable in performance to present-day supercomputers. The crossbar **switch** is comprised of two basic sections; a **multiplexer** and a control section. The **multiplexer** provides the actual **switching** of signal paths, i.e. connects each processing element unit to a particular parallel memory on each clock cycle (100 nsec). The control section determines which connections are made on each clock cycle in accordance with the algorithm under execution. Selectable pipelined delay in the control section provides for optimal data transfer efficiency between the processors and memory modules over a wide range of array processing algorithms. The crossbar **switch** also provides for graceful system degradation in computational throughput power without the need to download a new program.

Brief Summary Text - BSTX (8):

The configuration of the crossbar **switch**, (i.e. the connection between each floating point arithmetic PE unit and each PMEM module) may be changed dynamically on a cycle-by-cycle basis in accordance with the needs of the algorithm being executed. Although there are certain crossbar usage rules which must be obeyed, for a given algorithm, each PE unit can fetch from PMEM, operate on an input data stream and write back to PMEM in a highly parallel fashion producing a performance comparable to present-day supercomputers. A unique mapping of different data types (i.e. scalars, vectors and matrices) across PMEM in conjunction with the use of the proposed crossbar **switch** provides for very efficient and flexible data transfers between PEs and PMEMs that are free of communication bottlenecks. The crossbar **switch** is comprised of two basic sections: a **multiplexer** and a control section. The **multiplexer** provides the actual **switching** of signal paths, i.e., connects each PE unit to a particular PMEM on any clock cycle (100 nsec). The control section determines which connections are made on each clock cycle in accordance with the algorithm under execution. A selectable pipeline delay in the circuitry of the control section which produces a variety of interlaced patterns of memory reads and writes over a parallel bidirectional data bus provides for an optimal data transfer between the PEs and PMEM modules over a wide range of array processing algorithms. The highly parallel architecture of the present invention allows true supercomputer performance to be achieved with a clock rate which runs at a conventional PC rate of 100 ns per cycle. Thus, the supercomputing power of the present invention does not generate excessive heat during operation and may be air cooled and ruggedized quite readily.

Detailed Description Text - DETX (16):

A crossbar **switch** is a N.sup.2 interconnection network which allows N PEs to communicate simultaneously with N memories. The **switch** consists of two sections; a **multiplexer** and a control section.

Detailed Description Text - DETX (17):
The **multiplexer** section does the actual **switching** of signal paths. In general, there is a path from every device into the **switch**. The configuration of the **switch** determines which pairs of devices (i.e., which PMEM and PE) will be connected. In a true crossbar, all connections are possible.

Detailed Description Text - DETX (18):
The control section of the switch determines which connections are made at any one time. Each PE provides its own portion of the control field to the **crossbar** switch on every memory access cycle. The set of all of the PE control fields determines the configuration of the switch during any given memory access cycle. The **crossbar** switch configuration is, therefore, predetermined at the time that the PE microcode algorithms are developed. This approach eliminates the need for **arbitration** of access to PMEM's during run time, thereby allowing the **crossbar** switch to operate at or near 100% efficiency.

Claims Text - CLTX (2):
a **multiplexer** section for interconnecting said processing elements and memories during memory reads and memory writes by providing **switching** of signal paths between said processing elements and said memories; and

Claims Text - CLTX (3):
a control section coupled to said **multiplexer** section for controlling said signal path **switching** to allocate said signal paths in accordance with said steering vectors, said steering vectors being determined synchronously on each clock cycle by an algorithm under execution and thereby controlling the interconnection of said processing elements with said memories, said control section including control circuitry for creating pipeline delay during memory reads of said steering vectors and of data returning from said memories specified by said steering vectors.

Current US Original Classification - CCOR (1):
710/317

L Number	Hits	Search Text	DB	Time stamp
1	20	(((switch$3) same (crossbar$1 or crosspoint$1)) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/12/22 16:16
2	3	(simultaneous near3 access$3) and ((((switch$3) same (crossbar$1 or crosspoint$1)) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/12/22 16:13
3	4	(simultaneous$2 near3 transfer$5) and ((((switch$3) same (crossbar$1 or crosspoint$1)) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/12/22 16:13
4	4	((simultaneous$2 near3 transfer$5) and ((((switch$3) same (crossbar$1 or crosspoint$1)) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.)) not ((simultaneous near3 access$3) and ((((switch$3) same (crossbar$1 or crosspoint$1)) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/12/22 16:13
5	26	((crossbar$1 or crosspoint$1) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/12/22 16:17
6	6	(((crossbar$1 or crosspoint$1) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.) not ((((switch$3) same (crossbar$1 or crosspoint$1)) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/12/22 16:22
7	4	(multiplexer$1 same switch$3) and (((crossbar$1 or crosspoint$1) same (arbit$3 or arbitrat$3)) and 710/316,317.ccls.)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/12/22 16:23